

## IN THE CLAIMS:

---

1. (currently amended) A method in a data processing system for debugging a process from a starting point, comprising:
  - initiating debugging of a the process;
  - saving a process state in response to a first event to form a stored process state;
  - retrieving the stored process state in response to a ~~second~~ predefined event; and
  - reinitiating debugging from the stored process state.
2. (Original) The method of claim 1, wherein the first event occurs periodically.
3. (Original) The method of claim 1, wherein the process state is saved in a checkpoint data structure.
4. (Original) The method of claim 3, wherein the checkpoint data structure is a checkpoint file.
5. (Original) The method of claim 3, wherein the checkpoint data structure includes a process descriptor for the process.
6. (currently amended) [The method of claim 5] A method in a data processing system for debugging a process from a starting point, comprising:
  - initiating debugging of the process;
  - saving a process state in response to a first event to form a stored process state;
  - retrieving the stored process state in response to a second event; and
  - reinitiating debugging from the stored process state, wherein the process has control over at least one child process and the ~~checkpoint data structure~~ process state includes a process descriptor for each of the at least one child process.
7. (Original) The method of claim 5, wherein the checkpoint data structure further includes at least one data type descriptor for at least one data type corresponding to the process.

8. (Original) The method of claim 7, wherein the checkpoint data structure further includes at least one instance descriptor for at least one instance of data corresponding to each of the at least one data type.
9. (Original) The method of claim 8, wherein the checkpoint data structure further includes at least one data block corresponding to each of the at least one instance descriptor.
10. (Original) The method of claim 1, further comprising the step of modifying at least one register or memory variable before resuming debugging from the stored process state.
11. (Original) The method of claim 1, wherein the process state is saved when the program is in a stopped state.
12. (Original) The method of claim 11, wherein the stopped state is at a breakpoint.
13. (currently amended) An apparatus for debugging a process from a starting point, comprising:  
a processor; and  
a memory electrically connected to the processor, the memory having stored therein a program to be executed on the processor for performing the following steps:  
initiating debugging of a the process;  
saving a process state in response to a first event to form a saved process state;  
retrieving the saved process state in response to a ~~second~~  
predefined event; and  
reinitiating debugging from the saved process state.

14. (Original) The apparatus of claim 13, wherein the first event occurs periodically.
15. (Original) The apparatus of claim 13, wherein the process state is saved in a checkpoint data structure.
16. (Original) The apparatus of claim 15, wherein the checkpoint data structure is a checkpoint file.
17. (Original) The apparatus of claim 15, wherein the checkpoint data structure includes a process descriptor for the process.
18. (currently amended) [The apparatus of claim 17] An apparatus for debugging a process from a starting point, comprising:  
a processor; and  
a memory electrically connected to the processor, the memory having stored therein a program to be executed on the processor for performing the following steps:  
initiating debugging of a process;  
saving a process state in response to a first event to form a saved process state;  
retrieving the saved process state in response to a second event;  
and  
reinitiating debugging from the saved process state, wherein the process has control over at least one child process and the ~~checkpoint data structure~~ process state includes a process descriptor for each of the at least one child process.
19. (Original) The apparatus of claim 17, wherein the checkpoint data structure further includes at least one data type descriptor for at least one data type corresponding to the process.

20. (Original) The apparatus of claim 19, wherein the checkpoint data structure further includes at least one instance descriptor for at least one instance of data corresponding to each of the at least one data type.

21. (Original) The apparatus of claim 20, wherein the checkpoint data structure further includes at least one data block corresponding to each of the at least one instance descriptor.

A 22. (Original) The apparatus of claim 13, wherein program further comprises the step of modifying at least one register or memory variable before resuming debugging from the retrieved process state.

23. (Original) The apparatus of claim 13, wherein the process state is saved when the program is in a stopped state.

24. (Original) The apparatus of claim 23, wherein the stopped state is at a breakpoint.

25. (currently amended) A computer program product embodied in a computer readable medium for debugging a process from a starting point, comprising:  
instructions for initiating debugging of a the process;  
instructions for saving a process state in response to a first event to form a saved process state;  
instructions for retrieving the saved process state in response to a ~~second~~ predefined event; and  
instructions for reinitiating debugging from the saved process state.

---

A<sup>2</sup> 26. (newly added) The method of Claim 1, wherein the first event is a breakpoint and the predefined event is a checkpoint, and further comprising the step of repeatedly running between the checkpoint and the breakpoint for a plurality of times.

27. (newly added) The method of Claim 26 wherein variable values are automatically modified after retrieving the stored process state for the checkpoint.

28. (newly added) A method in a data processing system for debugging a process in a multi-process environment, comprising the steps of:

initiating a debug process;

creating a child process from the debug process;

saving a process state of the child process in response to a first event, to form a stored process state;

retrieving the stored process state in response to a second event; and

executing the child process using the stored process state.

29. (newly added) A method in a data processing system for debugging a process in a multi-process environment, comprising the steps of:

tracing a process by a debugger;

saving a process state of the traced process and a process state of another process that is not being traced by the debugger;

retrieving the saved process states; and

reinitiating debugging of the process using the retrieved process states.